

⑫ 公開特許公報(A)

昭60-163300

⑬ Int. Cl.

G 11 C 29/00
G 06 F 11/10
12/16

識別記号

庁内整理番号

7922-5B

7368-5B

7922-5B

⑭ 公開 昭和60年(1985)8月26日

※審査請求 未請求 発明の数 1 (全15頁)

⑮ 発明の名称 誤り訂正機能付半導体メモリ

⑯ 特 願 昭59-18326

⑰ 出 願 昭59(1984)2月6日

⑱ 発 明 者 堀 口 真 志 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑲ 発 明 者 下 東 勝 博 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑳ 発 明 者 青 木 正 和 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

㉑ 発 明 者 中 込 儀 延 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

㉒ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉓ 代 理 人 弁理士 高橋 明夫 外1名

最終頁に続く

明 細 書

発明の名称 誤り訂正機能付半導体メモリ

特許請求の範囲

1. q 通り($q \geq 3$)の互いに異なる量の電荷を蓄積することによつて q 値の情報を記憶するメモリセルを用い、誤り訂正符号として q 元符号を用いることを特徴とする、誤り訂正機能付半導体メモリ。
2. 上記 q 元符号として、 q 元巡回符号もしくは q 元短縮化巡回符号を用いることを特徴とする、特許請求の範囲第1項記載の誤り訂正機能付半導体メモリ。

発明の詳細な説明

〔発明の利用分野〕

本発明は、ソフトウェア対策として誤り訂正機能を設けたメモリに係り、特に1メモリセルに多値情報を記憶するメモリに関する。

〔発明の背景〕

キャパシタンスに蓄積する電荷量によつて情報を記憶するメモリセルには、1トランジスタ形メ

モリセルやCCDがある。このキャパシタンスに蓄積する電荷量を3通り以上にすることによつて1メモリセルに3値以上の情報を記憶する方式については、例えば文献Lewis M. Terman et. al., "CCD Memory using Multilevel Storage", ISSCC Tech. Dig. Papers, Fed. 1981, pp. 154-155に述べられている。このような多値記憶を実現する際に問題となるのは、 q 値等によつて生ずるソフトウェアである。2値情報の誤り対策としては、例えば文献Tsuneo Mano et. al., "Submicron VLSI Memory Circuits", ISSCC Tech. Dig. Papers, Feb. 1983, pp. 234-235に述べられているように検査ビットを設けて誤り訂正を行う方式がある。しかし、多値記憶の場合は、1個のメモリセルにソフトウェアが生じるとそれに記憶されている多値情報がすべて失われる(例えば8値記憶の場合は3ビットが同時に失われる)ため、通常の誤り訂正方式では訂正できない。このような誤りをも訂正できるようにする一つの方法は、多

重なり訂正符号を用いることである。例えば8値記憶の場合は三重なり訂正符号を用いればよい。しかし、一般に多重なりを訂正できるようにするためには、多数の検査ビットを必要とし、符号化、復号の手間も大きいという欠点がある。

〔発明の目的〕

本発明の目的は、1メモリセルに記憶されている多値情報がソフトウェアによつて一度に失われても、誤り訂正が容易な多値メモリを提供することにある。

〔発明の概要〕

上記のようなソフトウェアを効率よく訂正するためには、1メモリセルに記憶されている多値情報をまとめて1つのシンボルとみなし、そのシンボルを単位とした符号、すなわち多元符号を用いればよい。

〔発明の実施例〕

第1図に本発明の一実施例を示す。本実施例は、1トランジスタ形メモリセルに4値(2ビット)の情報を記憶させ、誤り訂正符号として4元符号

を用いる。

次に、メモリからデータを読み出す際の動作を説明する。データ書き込み時と同様に、読み出すべきアドレスに対応したワード線を1本(WL_i)とデータ線を5本(DL_j~DL_{j+4})を選択し、5個のメモリセルMC1_j~MC1_{j+4}を選択する。各メモリセルから読み出されたアナログ信号は、各データ線毎に設けられたAD変換器AD_j~AD_{j+4}によつて2ビットのデジタル値に変換される。メモリセル5個から読み出された計10ビットのデータ $a_{00}, a_{01}, \dots, a_{10}, a_{11}$ を復号回路2に入れて誤り訂正を行う。訂正されたデータ $a_{00}', a_{01}', \dots, a_{10}', a_{11}'$ は、DA変換器を通してもとのメモリセルに書き込むと同時に、6ビットのデータ $a_{10}', a_{11}', a_{20}', a_{21}', a_{30}', a_{31}'$ はデータ出力端子D_{out0}~D_{out5}から出す。

メモリセルのリフレッシュは、ワード線を1本選択し、各メモリセルから読み出されたアナログ信号を一旦AD変換器でデジタル信号に変換し

を用いたメモリである。以下、本実施例の動作を説明する。

まず、メモリにデータを書き込む際の動作を説明する。データ入力端子D_{in0}~D_{in5}から入つて来た6ビットのデータ $a_{00}, a_{01}, a_{10}, a_{11}, a_{20}, a_{21}$ を符号化回路1を通して4ビットの検査ビット $a_{00}, a_{01}, a_{10}, a_{11}$ を付加し、計10ビットの符号とする。一方、ワード線選択回路3およびデータ線選択回路4によつて、データを書き込むべきアドレスに対応したワード線を1本(WL_i)とデータ線を5本(DL_j~DL_{j+4})を選択し、5個のメモリセルMC1_j~MC1_{j+4}を選択する。書き込むべきデータは2ビットずつ組にして、 a_{00} と a_{01} とをメモリセルMC1_jに、 a_{10} と a_{11} とをMC1_{j+1}に、……、 a_{20} と a_{21} とをMC1_{j+4}に書き込む。そのため、各データ線毎に設けられたDA変換器DA_j~DA_{j+4}で2ビットの情報をアナログ電圧に変換し、その電圧をデータ線DL_j~DL_{j+4}を介してメモリセルMC1_j~MC1_{j+4}のキャパシタンスに蓄積すればよ

い。DA変換器で再びアナログ信号に戻してメモリセルに再書き込みすればよい。また、書き込み時、もしくは読出し時においても、選択されたワード線上にあつて選択されないデータ線にあるメモリセルについては、上記リフレッシュ動作を行う必要がある。

次に、この実施例に用いている符号化回路1、および復号回路2について詳細に説明する。第2図に符号化回路の回路図を、第3図に復号回路の回路図を示す。

まず、ここで用いている誤り訂正符号について述べる。符号化および復号の際には、同一のメモリセルに記憶する2ビット a_{i0} と a_{i1} ($i=0\sim4$)とをまとめて1つの4元のシンボルとして扱う。すなわち4元符号を用いる。したがつて、α線によつて1個のメモリセルがソフトウェア起因として2ビットの情報が同時に失われたとしても、他の4個のメモリセルがエラーを起こさなければ訂正することができる。

4元符号の4つのシンボルとしては、GF(4)

GF(4)は位数4の有限体の4個の元、0、1、 r 、 r^2 (ただし $r^2 + r + 1 = 0 \pmod{2}$) を用いる。2ビットのデータ (b_0, b_1) をこの4つのシンボルで表す際は、1と r との離形組合 $b_0 \cdot 1 + b_1 \cdot r$ で表す。すなわち、

$$0 \cdot 1 + 0 \cdot r = 0 \quad (1)$$

$$1 \cdot 1 + 0 \cdot r = 1 \quad (2)$$

$$0 \cdot 1 + 1 \cdot r = r \quad (3)$$

$$1 \cdot 1 + 1 \cdot r = 1 + r = r^2 \quad (4)$$

であるから、(0, 0)は0で、(1, 0)は1で、(0, 1)は r で、(1, 1)は r^2 で表す。

ここで用いている符号は4元ハミング(5, 3)符号であり、そのパリティ検査行列 \bar{H} は、

$$\bar{H} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & r & r^2 \end{pmatrix} \quad (5)$$

である。したがって、符号語を $\vec{a} = (a_0, a_1, a_2, a_3, a_4)$ とすると、 $\bar{H} \vec{a}^T = 0$ 、すなわち、

$$a_0 + a_2 + a_3 + a_4 = 0 \quad (6)$$

$$a_1 + a_2 + a_3 r + a_4 r^2 = 0 \quad (7)$$

であるから、21に示すようにEORゲート1個で実現できる。 a_4 と r^2 の積についても同様に22に示すようにEORゲート1個で実現できる。

次に、第3図の復号回路について説明する。復号回路は、シンδροームを計算する回路23と、誤り訂正を行う回路24から成る。

メモリから読み出された10ビット $a_{00}, a_{01}, \dots, a_{40}, a_{41}$ を、同一のメモリセルに記憶されていたもの同士2ビットずつまとめて、 $a_0 = a_{00} + a_{01} r$ 、 $a_1 = a_{10} + a_{11} r$ 、 $a_2 = a_{20} + a_{21} r$ 、 $a_3 = a_{30} + a_{31} r$ 、 $a_4 = a_{40} + a_{41} r$ の5個の4元シンボルとみなす。この受信系列 $\vec{a} = (a_0, a_1, a_2, a_3, a_4)$ から次の式に従ってシンδροーム $\vec{S} = \begin{pmatrix} S_0 \\ S_1 \end{pmatrix}$ を計算する。

$$\vec{S} = \bar{H} \vec{a}^T \quad (8)$$

すなわち、

$$S_0 = a_0 + a_2 + a_3 + a_4 \quad (9)$$

$$S_1 = a_1 + a_2 + a_3 r + a_4 r^2 \quad (10)$$

である。これを計算する回路は、符号化回路と同様に作るができる。

である。

次に、第2図の符号化回路について説明する。符号化回路では、データ入力端子から入って来た6ビット $a_{00}, a_{01}, a_{10}, a_{11}, a_{20}, a_{21}$ を同一のメモリセルに記憶するもの同士2ビットずつまとめて、前述のように $a_0 = a_{00} + a_{01} r$ 、 $a_1 = a_{10} + a_{11} r$ 、 $a_2 = a_{20} + a_{21} r$ の3個の4元シンボルとみなす。この3個を情報点として、 $a_3 = a_{00} + a_{01} r$ 、 $a_4 = a_{10} + a_{11} r$ の2個の検査点を、式(6)、(7)を満たすように付加して、符号語とする。そのためは、

$$a_3 = a_0 + a_2 + a_4 \quad (11)$$

$$a_4 = a_1 + a_2 r + a_3 r^2 \quad (12)$$

を計算すればよい。4元シンボル同士の加算は、排他的論理和 (exclusive OR, 以下EORと略す) ゲート2個で実現できる。また、 a_3 と r の積は、

$$\begin{aligned} a_3 \cdot r &= (a_{00} + a_{01} r) r = a_{00} r + a_{01} r^2 \\ &= a_{00} r + a_{01} (r + 1) \\ &= a_{01} + (a_{00} + a_{01}) r \end{aligned} \quad (13)$$

次に、このシンδροームを用いて誤りの生じた位置と誤りの大きさを決定し、訂正を行う。シンδροーム \vec{S} が、 $\bar{H} = (h_0, h_1, h_2, h_3, h_4)$ のある列ベクトル \vec{e}_j の c 倍に等しいとき、 a_j に大きさ c の誤りが生じたと判断し

$$\vec{a}' = \vec{a} + c \vec{e}_j \quad (14)$$

によって訂正された信号 \vec{a}' を作る。例えば a_3 が誤っているかどうかを調べるには、

$$\vec{S} = \begin{pmatrix} S_0 \\ S_1 \end{pmatrix} = c \begin{pmatrix} 1 \\ r \end{pmatrix} = \begin{pmatrix} c \\ c r \end{pmatrix} \quad (15)$$

を満たす c が存在するかどうか、すなわち、

$$S_1 = c r = S_0 r \quad (16)$$

が成り立つかどうかを調べればよい。成り立つ場合は、

$$\vec{a}' = \vec{a} + c = \vec{a} + S_0 \quad (17)$$

によって訂正を行い、成り立たない場合は a_j には誤りが生じなかったと判断して \vec{a} をそのまま \vec{a}' とすればよい。

第4図に本発明の他の実施例を示す。第1図との相違点は、データ入出力端子が各2ビットしか

ないことである。第1図では誤り訂正を行う1つのブロックに含まれる情報量とデータ入出力端子の数とはとも6ビットで等しいが、本実施例では異なるため、その動作は第1図の場合とはやや異なる。以下、本実施例の動作を説明する。

メモリからデータを読み出す際の動作は、第1図の場合とほとんど同じである。ただ、復号回路(復号回路は第3図と同じでよい)で誤りを訂正されたデータ6ビットのうち、2ビットを選択回路6で選択してデータ出力端子 D_{out0} 、 D_{out1} に出すだけである。

これに対して、メモリにデータを書き込む際は、選択されたメモリセルだけでなく、検査ビットが記憶されているメモリセルの内容をも書き替える必要があるため、その動作は第1図の場合とかなり異なる。まず、データ読み出し時と同様に、5個のメモリセル $MC11 \sim MC11.4$ から計10ビットのデータ a_{10} 、 a_{11} 、……、 a_{19} 、 a_{21} を読み出して、復号回路で誤り訂正を行う。この10ビットから検査ビットを除いた6ビット a'_{10} 、 a'_{11} 、

a'_{20} 、 a'_{21} 、 a'_{30} 、 a'_{31} をデータ置換回路5に入れる。ここでは、6ビットのうち2ビットをデータ入力端子 D_{in0} 、 D_{in1} から入つて来たデータで置換する(例えば図の状態では a'_{10} が D_{in0} で、 a'_{11} が D_{in1} でそれぞれ置換される)。この6ビットを符号化回路(符号化回路は第2図と同じでよい)に入れて検査ビットを付加し、もとのメモリセル $MC11 \sim MC11.4$ に2ビットずつ書き込むべきよい。

第5図に本発明の他の実施例を示す。第1図との相違点は、データ入出力端子が各1個しかなく、入出力をシリアルに行うことである。そのためシフトレジスタ7および8を設けてシリアル・パラレル変換を行っている。すなわち、メモリにデータを書き込む際は、データ入力端子 D_{in} から入つて来たデータを順にシフトレジスタ7に入れ、6ビット入れ終わつた後に符号化を行う。メモリからデータを読み出す際は、誤りを訂正されたデータを一旦シフトレジスタ8に入れ、順にデータ出力端子 D_{out} に出す。その他の動作は第1図の

場合と同様である。

第6図に本発明の他の実施例を示す。第1図との相違点は、第1図の場合はデータ線を5本同時に選択するが、本実施例では1本ずつ順次に選択し、メモリセルのデータ読み出し・書き込みをシリアルに行うことである。そのため2列5段の双方向シフトレジスタ9を用いてシリアル・パラレル変換を行っている。メモリにデータを書き込む際は、符号化回路1の出力を一旦シフトレジスタ9に入れ、9を右方向にシフトしながら5本のデータ線 $DL1.0$ 、 $DL1.1$ 、……、 $DL1$ を順に選択し、メモリセル $MC11.0$ 、 $MC11.1$ 、……、 $MC11$ の順に2ビットずつデータを書き込む。メモリからデータを読み出す際は、まずシフトレジスタ9を左方向にシフトしながらデータ線 $DL1$ 、 $DL1.1$ 、……、 $DL1.4$ を順に選択し、メモリセル $MC11$ 、 $MC11.1$ 、……、 $MC11.4$ の順にデータを読み出す。次に、復号回路2を動作させて誤りを訂正し、訂正したデータは再びシフトレジスタ9に書き込むと同時に6ビットはデータ出力端

子 $D_{out0} \sim D_{out4}$ に出す。最後に、シフトレジスタ9を右方向にシフトしながらデータ線 $DL1.4$ 、 $DL1.3$ 、……、 $DL1$ を順に選択し、メモリセル $MC11.4$ 、 $MC11.3$ 、……、 $MC11$ の順にデータの書き込みを行う。

なお、本実施例では、データ線を1本ずつ順次に選択するが、第7図に示すようにデータ線は5本まとめて($DL1 \sim DL1.4$)選択し、そのかわりにシフトレジスタ10を設けてもよい。

第6図および第7図に示した実施例では、データの入出力は第1図と同様に6ビット並列に行っているが、第4図もしくは第5図に示したようなデータ入出力の方法を採用してもよい。

第8図に本発明の他の実施例を示す。本実施例はいわゆるブロックオリエンテッドRAM(以下BORAMと略す)であり、1本のワード線に接続されているすべてのメモリセルを1つのブロックとして、ブロック単位に読み出し・書き込みを行うメモリである。図の例では、1本のワード線に5個のメモリセルが接続され、1メモリセル

には4値の情報が記憶されているので、1ブロックの大きさは検査ビットを含めて10dビット、検査ビットを除くと6dビットである。以下、本実施例の動作を説明する。

メモリのデータ読み出し・書き込みは、リング状に接続されたシフトレジスタ11および12を介して行う。2列5d段のシフトレジスタ11はデータ線とのデータの授受に使用し、2列5d段のシフトレジスタ12は符号化回路1および復号回路2とのデータの授受に使用する。

メモリにデータを書き込む際は、まずデータ入力端子から入って来たデータをシリアルにシフトレジスタ7に入れる。データが6ビット入つて来るとに符号化回路1(符号化回路は図2図と同じでよい)を動作させ、検査ビット4ビットを付加してシフトレジスタ12に入れる。次にシフトレジスタ11および12をシフトして12の中に入っているデータを11に移す(これはD10から次のデータを入れるのと同じに行つてよい)。すべてのデータ(計10dビット)をシフトレジ

スタ11に移し終わつたところでメモリアルMC10~MC10i-1に2ビットずつデータを書き込む。

メモリからデータを読み出す際は、まず各データ線から読み出された計10dビットのデータをシフトレジスタ11に入れる。次に、シフトレジスタ11および12をシフトして11に入っているデータを12に移す。データを10ビット移すとに(5回シフトするとに)復号回路2(復号回路は図3図と同じでよい)を動作させて誤り訂正を行う。訂正されたデータは再びシフトレジスタ12に入れると同時に、6ビットはシフトレジスタ8に入れる。次に、シフトレジスタ11および12をシフトして次のデータを12に移すと同時に、訂正の終わつたデータを11に戻す。同時にシフトレジスタ8をシフトしてデータを出方端子Doutに出す。すべてのデータ(計10dビット)を訂正してシフトレジスタ11に戻し終わつたところでメモリアルMC10~MC10i-1にデータの再書き込みを行う。

第9図に本発明の他の実施例を示す。本実施例

も第8図と同様BOBAMであるが、相違点はAD変換器およびDA変換器を各データ線毎に設けず、シフトレジスタ12の前段に設けたことである。各データ線とAD変換器、DA変換器との間のデータの転送は、CCD13によつてアナログデータのままで行う。その他の動作は第8図と同様である。

第10図に本発明の他の実施例を示す。第9図の実施例はアナログ情報の転送にCCDを用いた例であるが、本実施例はCCD自体をメモリアルとして用い、これに多値情報を記憶させる方式のメモリである。本実施例の動作は、第9図の場合における1トランジスタ形メモリアルとCCDとの間のデータ転送が不要だけで、その他は第9図と同様である。

以上の実施例はいずれも符号化回路および復号回路としてそれぞれ第2図および第3図の回路を使用していたが、符号化回路、復号回路はこれに限らない。第11図および第12図にそれぞれ符号化回路、復号回路の他の実施例を示す。第2図

および第3図の実施例では並列に符号化および復号を行うのに対し、本実施例では符号として巡回符号を用い、その性質を利用してシリアルに符号化および復号を行う。

まず、ここで用いている誤り訂正符号について述べる。この符号でも、第2図および第3図の場合と同様に、同一のメモリアルに記憶する2ビット s_{i0} と s_{i1} ($i=0\sim4$)とをまとめて1つの4元シンボル $s_i = s_{i0} + s_{i1}r$ とみなす。この符号は4元ヘミング(5, 3)符号であり、そのパリタイ検査行列 H は、

$$\vec{H} = \begin{pmatrix} 1 & 0 & 1 & r & r \\ 0 & 1 & r & r & 1 \end{pmatrix} \quad (8)$$

である。これは

$$G(x) = x^5 + rx + 1 \quad (9)$$

を生成多項式とする巡回符号である。すなわち、符号語(s_0, s_1, s_2, s_3, s_4)を係数とするGF(4)上の多項式

$$P(x) = s_0 + s_1x + s_2x^2 + s_3x^3 + s_4x^4 \quad (10)$$

はG(x)で割り切れるという性質がある。

Cの性質を利用して3個の情報点 a_1, a_2, a_3 に検査点 a_0, a_1 を付加するには次のようにすればよい。まず、 a_1, a_2, a_3 を係数とする多項式

$$A(x) = a_1 x^2 + a_2 x + a_3 \quad (20)$$

を作る。 $A(x)$ を $G(x)$ で割った剰余を

$$B(x) = a_0 + a_1 x \quad (21)$$

とすると、 $A(x) + R(x)$ は $G(x)$ で割り切れるから、 $R(x)$ の係数 a_0, a_1 を検査点とすればよい。

第11図は以上述べた演算を行う回路である。4個のDフリップフロップ $FF_{10}, FF_{11}, FF_{12}, FF_{13}$ は共通のクロックによつて駆動され、2個の4元シンボル b_0, b_1 を記憶する役割を果たす。すなわち、 FF_{11} の出力を b_{11} とすると、

$$b_0 = b_{00} + b_{01} r \quad (22)$$

$$b_1 = b_{10} + b_{11} r \quad (23)$$

である。スイッチ用信号 SW_1 を“1”にして入力端子 I_0, I_1 にそれぞれ C_0, C_1 (4元シンボル $C = C_0 + C_1 r$ とみなす)を入れてクロックを印加すると、回路の状態は次のように変化する。

る。最後に、スイッチ用信号 SW_1 を“0”にしてスイッチ SW_2 を上側に倒し、クロックを2回印加して(このとき入力端子は“0”にしておく)回路の中に記憶されている a_1, a_2 をシフトして出力端子に取り出せばよい。

この符号の復号は次のようにして行えばよい。まず、メモリから読み出されたデータ $\vec{a} = (a_0, a_1, a_2, a_3)$ からシンδροーム $\vec{S} = \begin{pmatrix} s_0 \\ s_1 \end{pmatrix} = \vec{H} \vec{a}^T$ を求める。

$$s_0 = a_0 + a_2 + a_3 r + a_4 r \quad (24)$$

$$s_1 = a_1 + a_3 r + a_4 r + a_5 \quad (25)$$

であるから、

$$\begin{aligned} B(x) &= a_0 + a_1 x \\ &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \\ &\quad + G(x) (a_2 + (x+r)a_3 + (x^2+rx+r)a_4) \end{aligned} \quad (31)$$

となる。したがって、 B の要素を係数とする多項式

$$B(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \quad (32)$$

$$b_0^{(n+1)} = b_1^{(n)} + C \quad (33)$$

$$b_1^{(n+1)} = b_0^{(n)} + r(b_1^{(n)} + C) \quad (34)$$

ただし、上フヤの添字 (n) はクロックを n 回印加した後の状態であることを示す。したがって、 b_0, b_1 を係数とする多項式 $B(x) = b_0 + b_1 x$ は次のように変化する。

$$\begin{aligned} B(x)^{(n+1)} &= b_1^{(n)} + C + (b_1^{(n)} + r b_1^{(n)} + r C) x \\ &= (B(x)^{(n)} + C x) x + G(x) (b_1^{(n)} + C) \end{aligned} \quad (35)$$

結局、 $B(x)$ に Cx を加えて x を乗じ、生成多項式 $G(x)$ で割った剰余が新しい $B(x)$ となる。

符号化は次のような手順で行う。まず、すべてのフリップフロップを“0”にリセットする。次に、スイッチ用信号 SW_1 を“1”にしてスイッチ SW_2 を下側に倒し、クロックを印加しながら入力端子 I_0, I_1 から a_1, a_2, a_3 を順に入力する。このとき出力端子には、 a_1, a_2, a_3 がそのまゝ出て来る。回路の中では上に述べた演算が3回行われ、その結果 $A(x) = a_1 x^2 + a_2 x + a_3$ を $G(x)$ で割った剰余 $B(x) = a_0 + a_1 x$ が求められ

を生成多項式 $G(x)$ で割った剰余を求めれば、その係数がシンδροームとなる。

次に、このシンδροームを用いて誤りの生じた位置と誤りの大きさを決定し、訂正を行う。メモリから読み出されたデータのうち、 a_1 に大きさ e の誤りが生じているとすると、

$$S(x) = e x^j + Q(x) G(x) \quad (36)$$

と表される($Q(x)$ は多項式)。したがって、

$$\begin{aligned} S(x) x^{5-j} &= e x^5 + Q(x) G(x) x^{5-j} \\ &= e + (Q(x) x^{5-j} + x^5 + r x^3 + r x + 1) G(x) \end{aligned} \quad (37)$$

であるから、 $S(x)$ に $(5-j)$ 回 x を乗じて $G(x)$ で割った剰余が定数項 e のみになつたとき、 a_1 に大きさ e の誤りが生じていると判断して、

$$a_1' = a_1 + e \quad (38)$$

によつて訂正された信号 a_1' を作ればよい。

第12図はこの演算を行う回路である。第11図の場合と同様に x を乗じて $G(x)$ で割った剰余を求める回路を用いている。

復号は次のような手順で行う。まず、すべての

フリップフロップを“0”にリセットする。次に、クロックを印加しながら入力端子 I_0, I_1 から a_0, a_1, a_2, a_3, a_4 を順に入力する。同時に $a_0 \sim a_4$ はシフトレジスタ 25 (フリップフロップ $FF_{00} \sim FF_{11}$ と同じクロックで駆動される) に書込んでいく。このとき回路の中では、 $a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$ を $G(x)$ で割った剰余 $a_5 + a_1 x$ が求められる。次に、 I_0, I_1 を“0”にしてさらにクロックを印加し、 x を乗じて $G(x)$ で割った剰余を求める演算を繰り返す。この演算を $(5-i)$ 回行つたとき、結果が定数項のみになつたとすると、NORゲート 26 の出力が“1”になり、そのときシフトレジスタ 25 から出て来た信号 a_i が訂正される。

なお、符号化回路と復号回路とは共通部分が多いので第 13 図に示すように一つにまとめることも可能である。

第 11 図の符号化回路、第 12 図の復号回路、および第 13 図の符号化・復号回路では、データの入出力を 2 ビットずつシリアルに行うため、こ

れらを用いる場合はメモリの構成を多少変更する必要がある。例えば、第 1 図に示すメモリに第 11 図の符号化回路および第 12 図の復号回路を用いる場合は、第 14 図に示すように、シリアル・パラレル変換のためにシフトレジスタ 15, 16, 17 を付加する必要がある。また、第 8 図に示すメモリに適用する場合は、第 15 図に示すようにシフトレジスタ 12 を除去し、シフトレジスタ 7, 8 をそれぞれ 18, 19 で置き換えればよい。

以上の実施例はいずれも式 (5) もしくは (8) をパリタイ検査行列とする 4 元 (5, 3) 符号を用いた例であつたが他の符号でもよいことはもちろんである。例えば、

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & r & r & r & r & r & r & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & r & r & 0 & 1 & r & r & 0 & 1 & r & r & 0 & 1 & r & r & r \end{pmatrix} \quad (36)$$

をパリタイ検査行列とする 4 元 (21, 18) 符号でもよい。また、メモリセル 1 個に記憶する情報量は 4 値に限らない。一般に、 q 値 ($2 \leq q$

ビット) の情報を記憶する方式では、誤り訂正符号として q 元符号を用いる。一例として $q=8$ の場合について述べる。

8 元シンボルとしては $GF(8)$ の 8 個の元、0, 1, β , β^2 , ..., β^6 ($\beta^3 + \beta + 1 = 0 \pmod{2}$) を用いる。誤り訂正符号としては、例えば

$$H = \begin{pmatrix} 1 & 0 & 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 \\ 0 & 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 & 1 \end{pmatrix} \quad (37)$$

をパリタイ検査行列とする 8 元 (9, 7) 符号がある。これは、

$$G(x) = x^2 + \beta x + 1 \quad (38)$$

を生成多項式とする巡回符号である。この符号による誤り訂正機能を設けた実施例を第 16 図に示す (これは第 15 図と同様な構成の BOBAM である)。この実施例に用いている符号化回路および復号回路の回路図をそれぞれ第 17 図および第 18 図に示す (これらはそれぞれ第 11 図および第 12 図と同様、巡回符号の性質を利用した回路である)。

以上の例はいずれも、誤り訂正符号としては単

一誤り訂正符号を用いているが、単一誤り訂正二重誤り検出符号、あるいは多重誤り訂正符号を用いてもよいことはもちろんである。

〔発明の効果〕

以上説明したように、本発明によるメモリでは 1 個のメモリセルに記憶される q 値 ($q \geq 3$) の情報をまとめて 1 つの q 元シンボルとみなし、このシンボルを単位として符号化、復号を行う。したがつて、 q 値によつて 1 個のメモリセルに記憶されている q 値の情報がすべて失われるという種のソフトウェアの修正を容易に行うことができる。図面の簡単な説明

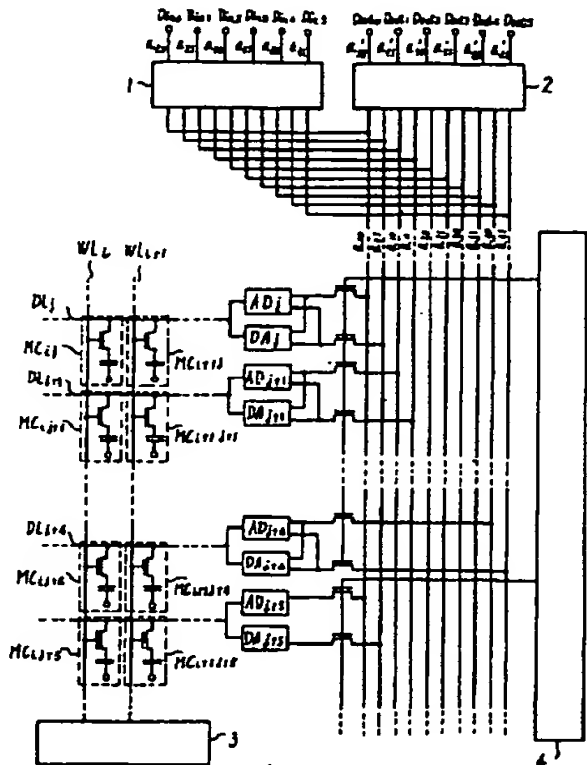
第 1 図、第 4 図～第 10 図、第 14 図～第 16 図は、本発明による誤り訂正機能付メモリの構成図、第 2 図、第 11 図、第 17 図は上記メモリに用いる符号化回路の回路図、第 3 図、第 12 図、第 18 図は上記メモリに用いる復号回路の回路図、第 13 図は上記メモリに用いる符号化・復号回路の回路図である。

1…符号化回路、2…復号回路、3…ワード経過

択回路、4…データ線選択回路、5…データ置換回路、6…選択回路、7, 8, 11, 12, 15, 16, 17, 18, 19, 25…シフトレジスタ、9, 10…双方向シフトレジスタ、13, 14…CCD、21… r を乗ずる回路、22… r^2 を乗ずる回路、23…シンドローム計算回路、24…訂正回路、26…NORゲート、27… β を乗ずる回路、MC11…メモリセル、WL1…ワード線、DL1…データ線、AD1…AD変換器、DA1…DA変換器、FF1…Dフリップフロップ。

代理人 弁理士 高橋明夫

第 1 図



第 3 図

第 2 図

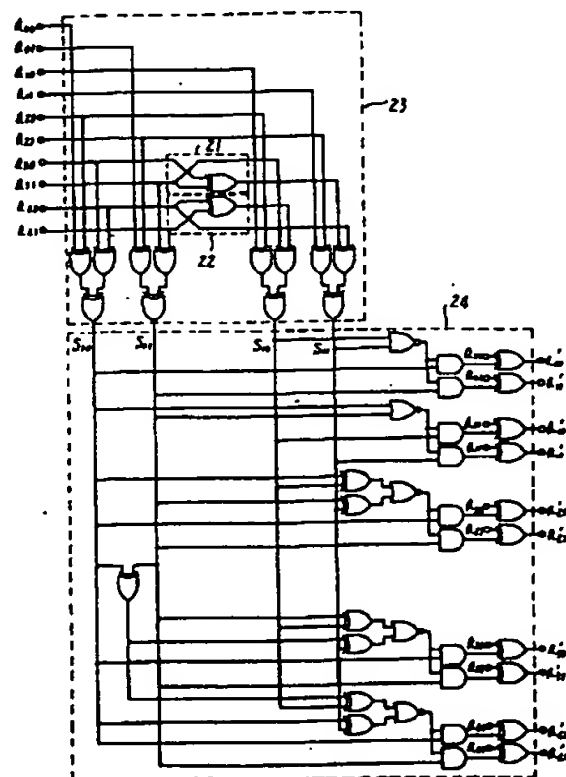
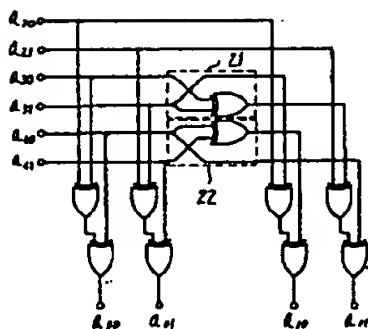


図 4

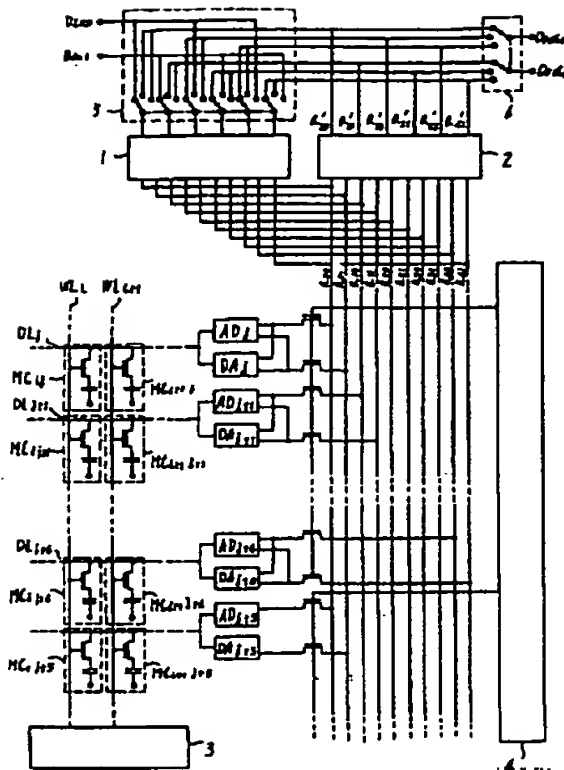


図 5

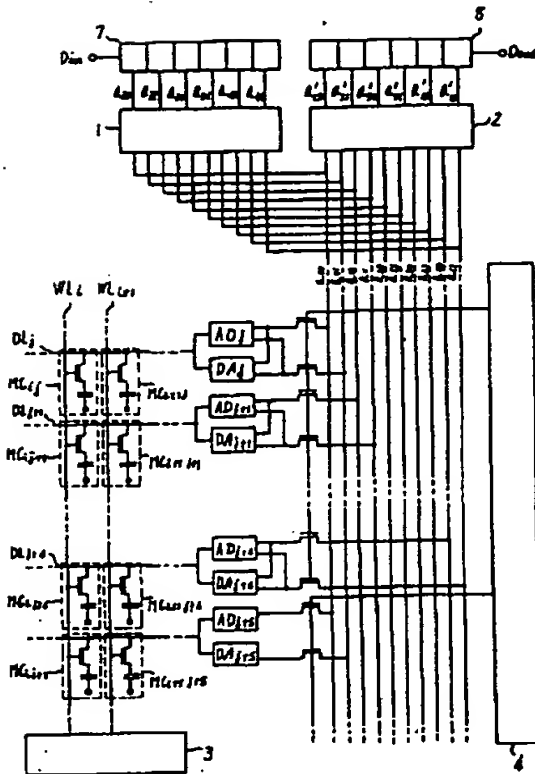


図 6

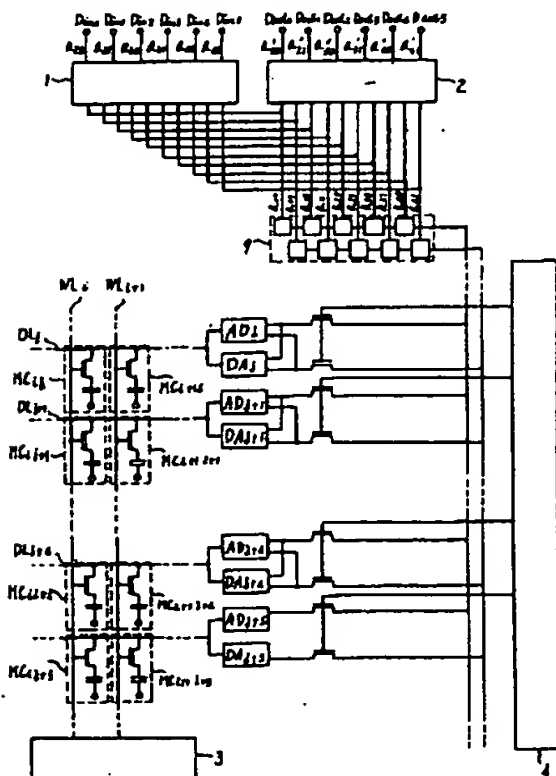


図 7

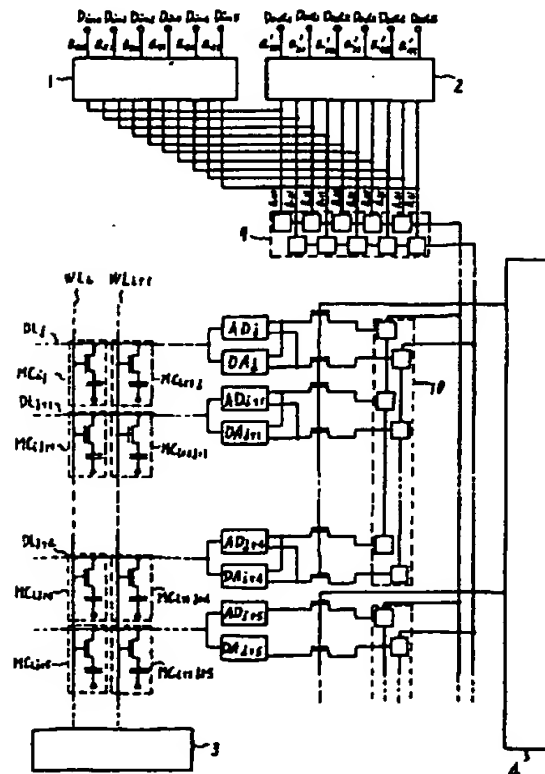


図 8

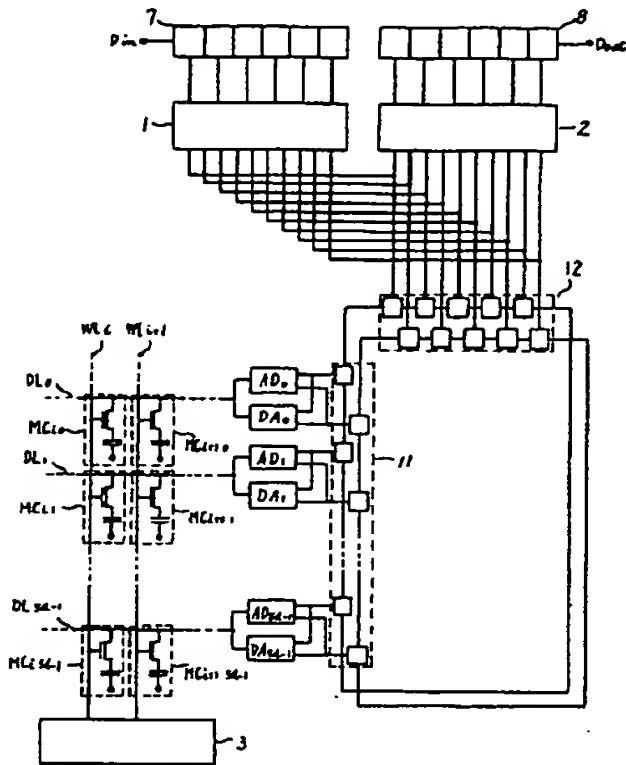


図 9

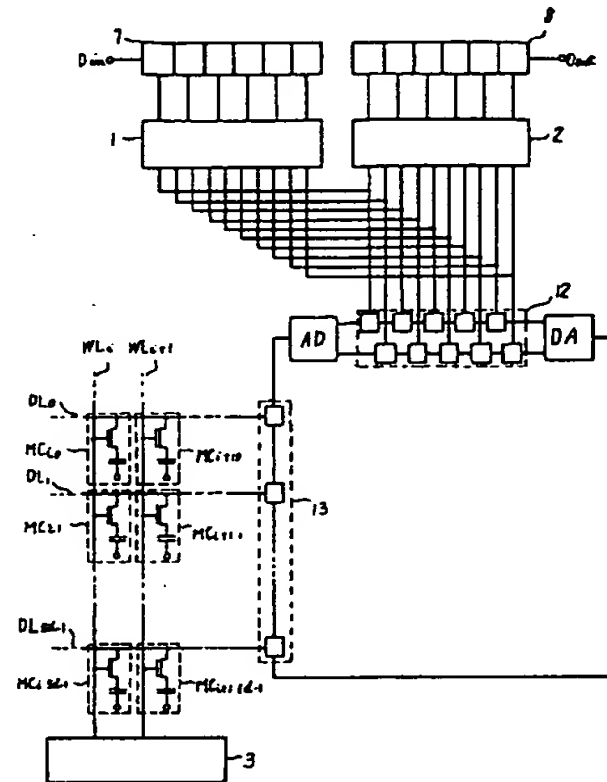


図 10

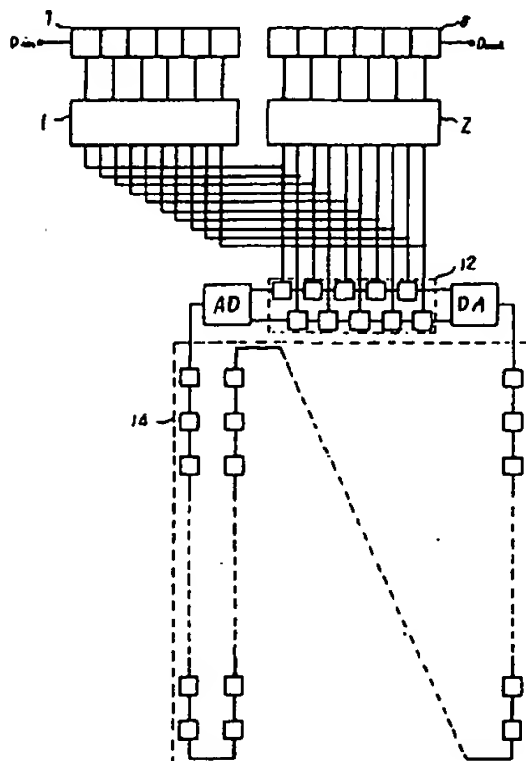
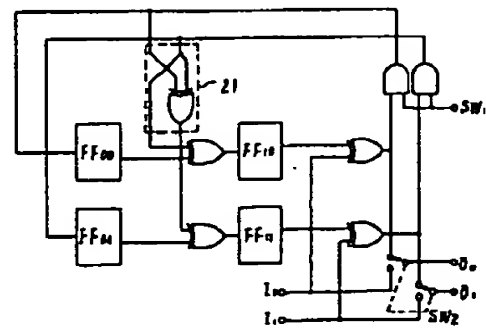
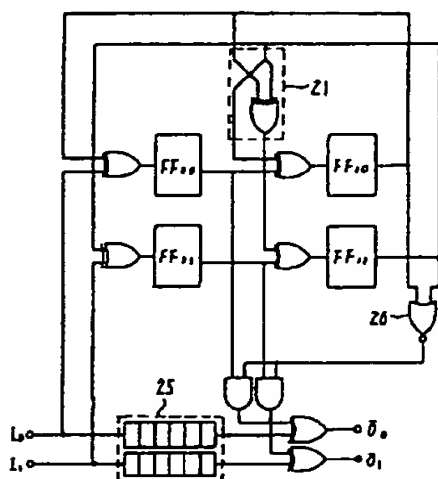


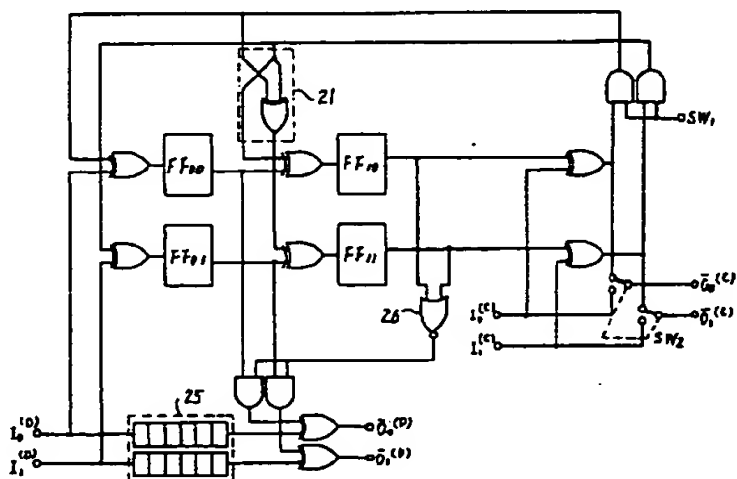
図 11



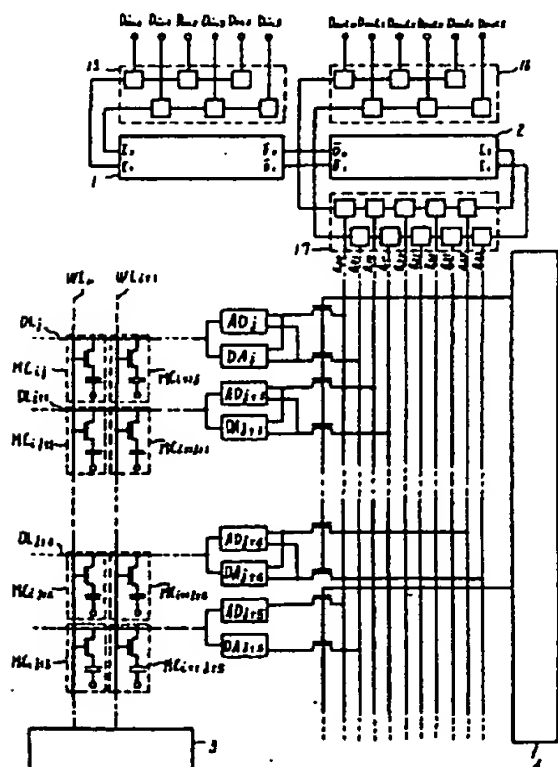
第 12 図



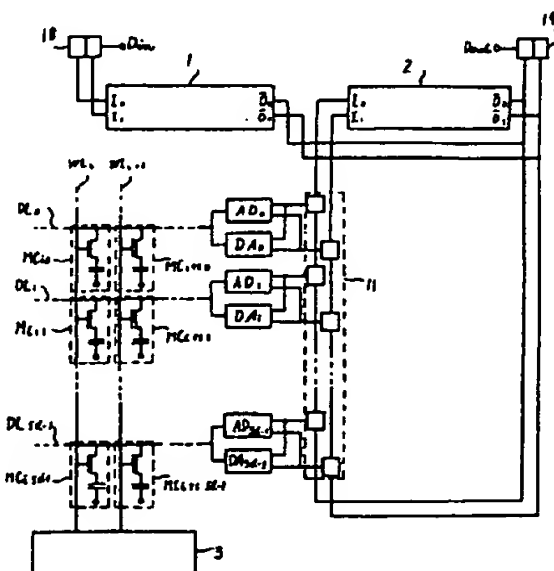
第 13 図



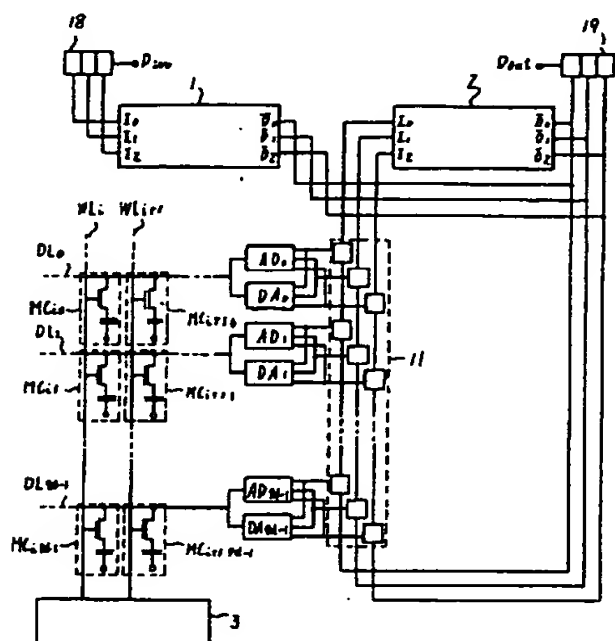
第 14 図



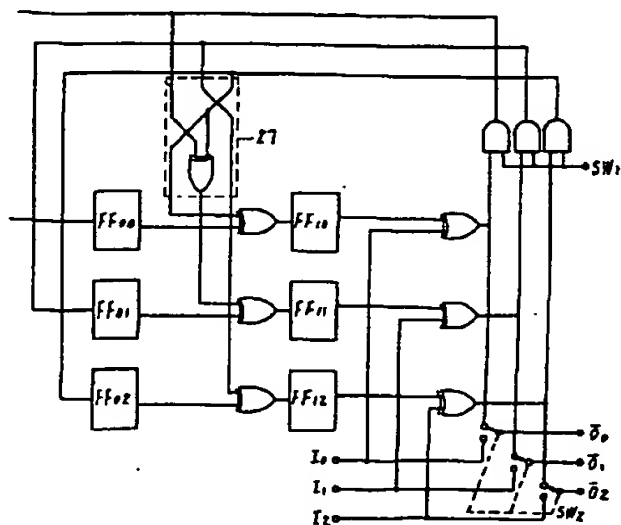
第 15 図



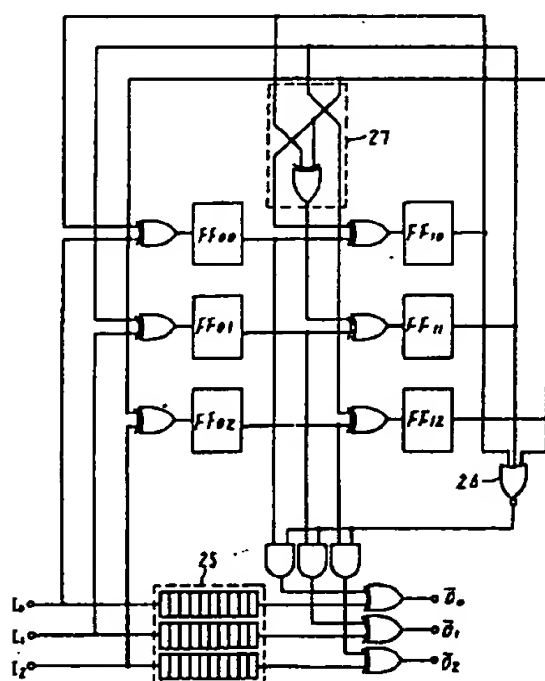
第 16 図



第 17 図



第 18 図



第1頁の続き

④Int. Cl.

G 11 C 11/56

識別記号

庁内整理番号

8219-5B

⑥発明者 池 永 伸 一 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

手続補正書

昭和59年3月30日

特許庁長官殿
事件の表示

昭和59年特許願第18326号
昭和59年2月6日付提出の特許願(26)
発明の名称

誤り訂正機能付半導体メモリ

補正をする者

特許出願人

株式会社日立製作所

補正の内容

1. 図面第1図、第4図から第7図と第14図を別紙のとおりに補正する。
2. 明細書について下記の補正をする。
 - (1) 第2頁第6行の「Fed.」を「Feb.」と訂正する。
 - (2) 第6頁第3行の「漏洩された」を「選択された」に訂正する。
 - (3) 第8頁第15行の「exclusive」を「exclusive」に訂正する。
 - (4) 第11頁第10行の「出力端子」を「出力端子」に訂正する。

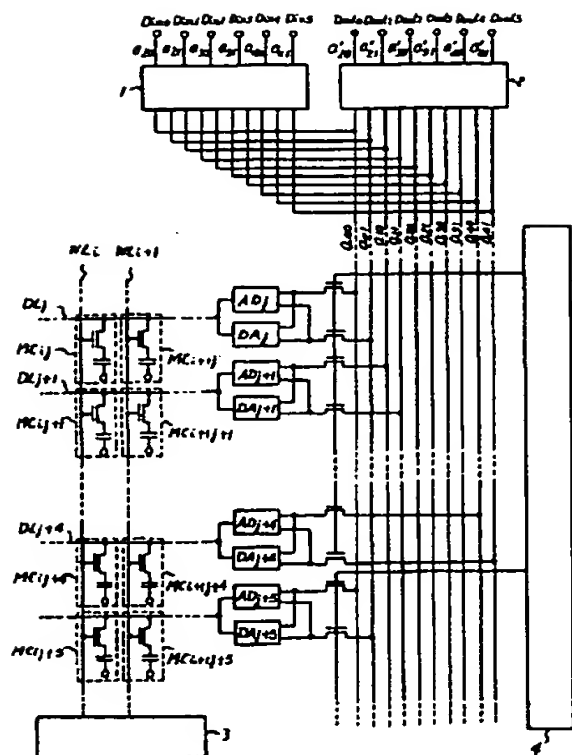
代理人

〒100 東京都千代田区丸の内一丁目5番1号
株式会社日立製作所内 電話 東京 212-1111(代表)
氏名 (6199) 高橋 明 夫

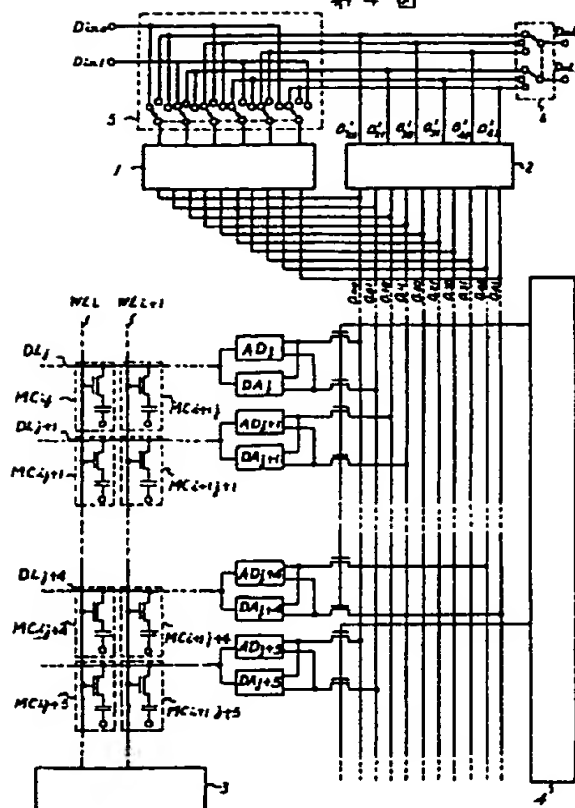
補正の対象 明細書の「発明の詳細な説明」の欄

補正の内容

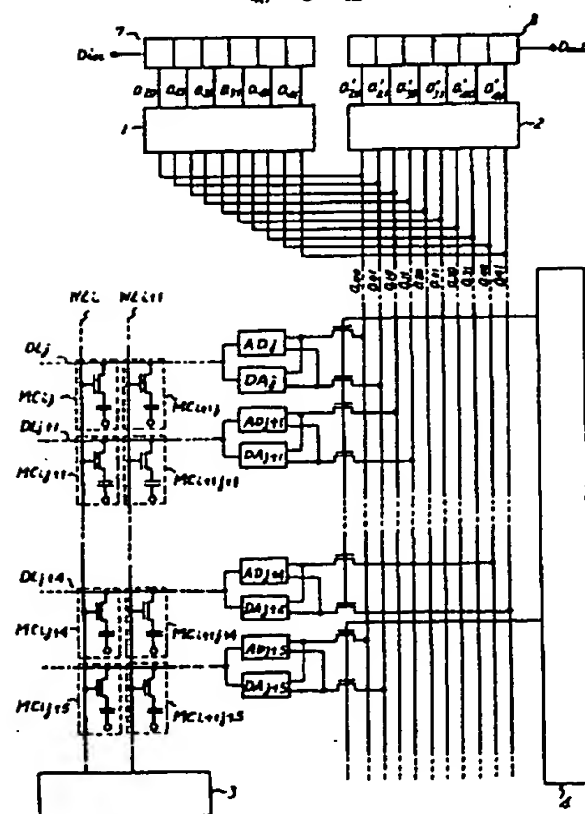
第 1 图



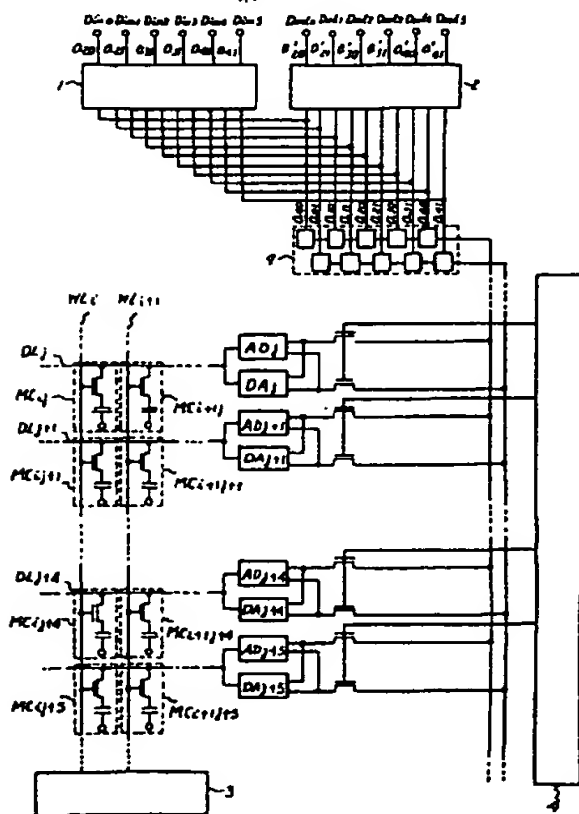
第 4 图



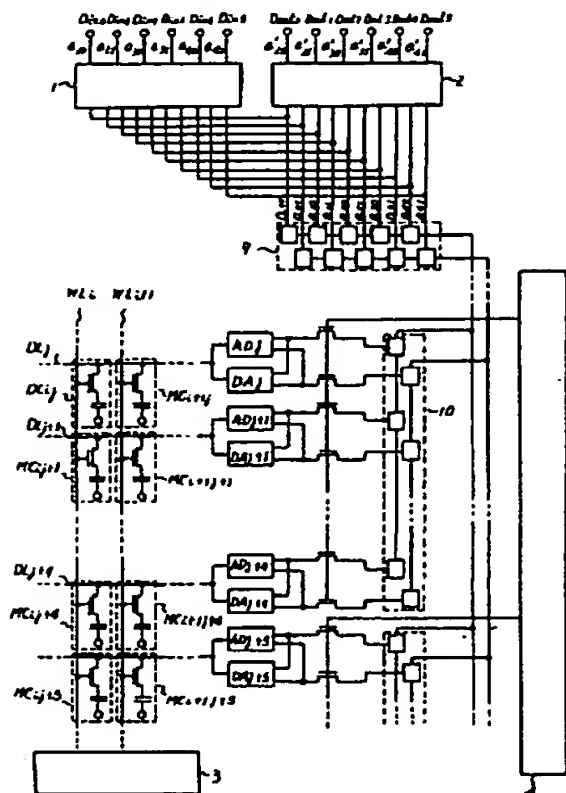
第 5 图



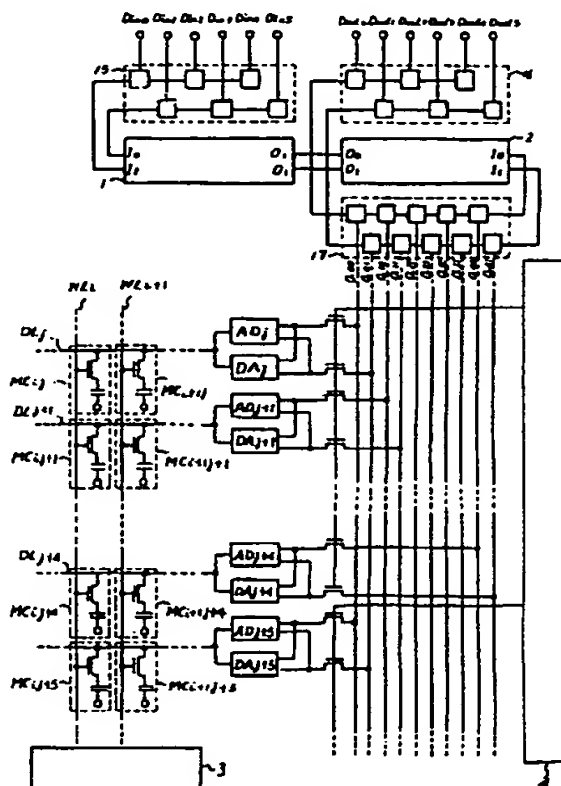
第 6 图



第 7 図



第 14 図



特許法第17条の2の規定による補正の掲載

平成 3. 5. 30 発行

昭和 59 年特許願第 18326 号 (特開昭
60-163300 号, 昭和 60 年 3 月 26 日
発行 公開特許公報 60-1633 号掲載) につ
いては特許法第17条の2の規定による補正があっ
たので下記のとおり掲載する。 6 (1)

Int. Cl. ¹	識別 記号	庁内整理番号
G11C 29/00		7737-58
G06F 11/10		9072-58
12/16		7737-58
G11C 11/56		7131-58

平成 3. 5. 30 発行

手 続 補 正 書

平成 3 年 1 月 30 日

特許庁長官 殿

事件の表示 昭和 59 年特許願第 18326 号

発明の名称 誤り訂正機能付半導体メモリ

補正をする者

事件との関係 特許出願人

名称 (510) 株式会社 日立製作所

代理人

居所 〒100 東京都千代田区丸ノ内一丁目5番1号
株式会社 日立製作所内
電話 東京 J212-1111 (大代表)

氏名 (6850) 弁理士 小川 勝男

補正の対象 明細書の「特許請求の範囲」の個

補正の内容

特許請求の範囲を別紙のとおり補正する。



特許請求の範囲

1. q 値の情報を記憶する誤り訂正機能付半導体メモリにおいて、

誤り訂正手段と、データ入力手段と、データ出力手段とを有し、

q は 3 以上であり、

誤り訂正符号として q 元符号を用いることを特徴とする誤り訂正機能付半導体メモリ。

2. 特許請求の範囲第 1 項記載の誤り訂正機能付半導体メモリにおいて、

上記 q 元符号は、q 元巡回符号であることを特徴とする誤り訂正機能付半導体メモリ。

3. 特許請求の範囲第 1 項記載の誤り訂正機能付半導体メモリにおいて、

上記 q 元符号は、q 元短縮化巡回符号であることを特徴とする誤り訂正機能付半導体メモリ。

4. 特許請求の範囲第 1 項記載の誤り訂正機能付半導体メモリにおいて、

上記誤り訂正手段は、符号化回路と、復号回路とを有し、

上記符号化回路は、入力データを上記 q 元符号に変換する機能を有し、

上記復号回路は、上記 q 元符号を出力データに変換する機能を有することを特徴とする誤り訂正機能付半導体メモリ。

5. 特許請求の範囲第 4 項記載の誤り訂正機能付半導体メモリにおいて、

上記出力データは、パラレルデータであることを特徴とする誤り訂正機能付半導体メモリ。

6. 特許請求の範囲第 4 項記載の誤り訂正機能付半導体メモリにおいて、

上記出力データは、シリアルデータであることを特徴とする誤り訂正機能付半導体メモリ。

7. 特許請求の範囲第 6 項記載の誤り訂正機能付半導体メモリにおいて、

上記データ出力手段は、シフトレジスタを有し、

上記シフトレジスタは、パラレルデータを上記シリアルデータに変換することを特徴とする誤り訂正機能付半導体メモリ。